# Deep Fake Face Detection: Pairwise Learning Approach

## Introduction

Deepfake technology has gained increasing attention due to the ongoing research in generating more believable fake images and detecting real vs fake images. Two encoder-decoder pairs have been a ubiquitous architecture for generating deepfakes. As research has progressed, numerous GAN architectures have emerged, each with their own unique features for producing fake images.

## Objectives

- Explore the detection of face images generated by various GAN architectures.
- Address the challenges faced by existing models such as the inability to generalize detection across GAN architectures that the model was not trained on, and the absence of statistical analysis between real and fake images.
- Implement and improve the performance of the pairwise learning model and gain a deeper understanding of its learning through network visualization.

## Model Design: Brief Overview of CFFN and CN

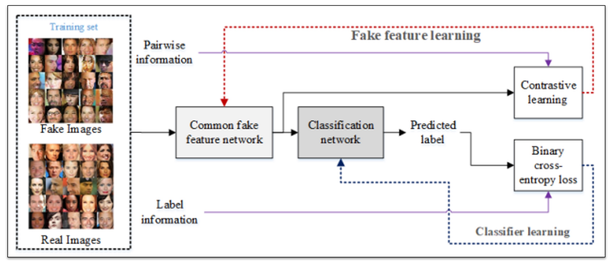


Figure 1: CFFN + CN architecture from "Deep Fake Image Detection Based on Pairwise Learning"

### Common Fake Feature Network (CFFN):

- Training Purpose: To learn the discriminative features of pairs of (real, real) & (fake, fake) images.
- Network Architecture: Acknowledgement - Dense net layers were implemented from Huang G., et. al's "Densely Connected Convolutional Networks."

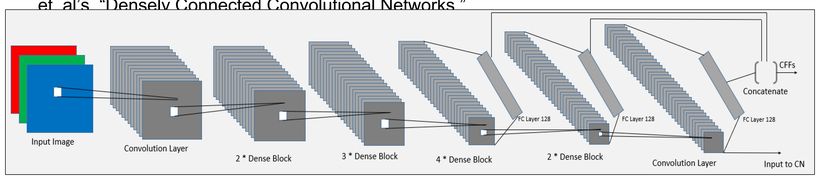


Figure 2: CFFN architecture

- Contrastive Loss Function:
- Pairwise label "y", where y = 0 indicates a fake pair, and y = 1 indicates a genuine pair.
- Minimizing the Energy Function - Constant mapping problem and use of contrastive loss.
- Constraints effect on Contrastive loss - purpose of "m" value and max function implementation.

$$E_w(x_1, x_2) = \|f_{CFFN}(x_1) - f_{CFFN}(x_2)\|_2^2$$

$$L(W, (P, x_1, x_2)) = 0.5 \times (y_{ij} E_w^2) + (1 - y_{ij}) \times \max(0, (m - E_w)_2^2)$$

Equation 1: CFFN Loss Function

### Classification Network (CN):

- Training Purpose: To distinguish between real and fake images in a binary classification task.
- Network Architecture: Convolutional layer with two channels and a FCL with two neurons.
- Two-Step Learning: (Pairwise Learning Strategy + Classifier Learning)

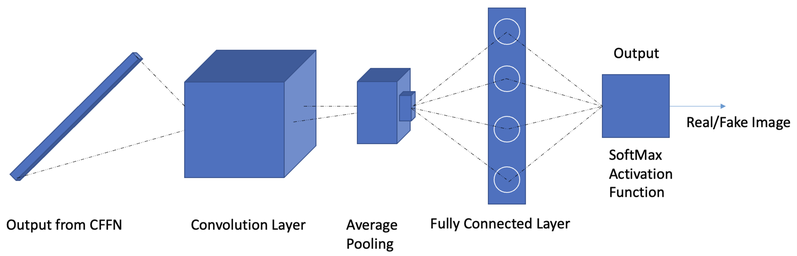


Figure 3: CN architecture

## Setup: Initial Experiments (Phase One)

### Dataset Generation:

- DCGAN, WGAN, and PGGAN models were used on CelebA dataset to generate fake images.
- 40,000 fake images were produced per GAN, resulting in a total of 120,000 fake images.
- To avoid data imbalance, equal number of real and fake 64x64 RGB images were considered.

### Training CFFN Model:

- Used 2 million real-real and fake-fake image pairs for 15 epochs with Batch size of 88.
- Included fake-fake image pairs generated by GANs for improved feature learning which helps the CFFN learn the common features between similar images.

### Training CN Model:

- Trained for 25 epochs with all real and fake images and batch size of 88.
- Used the output from the last convolutional layer of the CFFN as input to to CN.

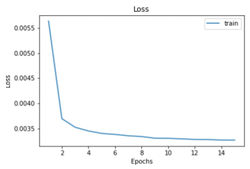


Figure 4: Training loss of CFFN over 15 epochs

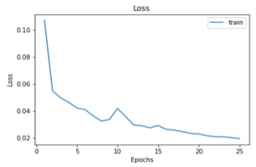


Figure 5: Training loss of CFFN over 25 epochs

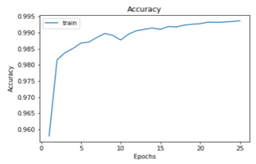


Figure 6: Accuracy of CN over 25 epochs ~ 99.5%

| *Epochs* | true_positive | false_positive | true_negative | false_negative | accuracy | recall | precision | f1_score |
|---|---|---|---|---|---|---|---|---|
| 15 | 939775 | 19081 | 957320 | 3632 | 0.9919 | 0.9962 | 0.9801 | 0.9881 |
| 16 | 1056407 | 22306 | 1072104 | 8967 | 0.9918 | 0.9916 | 0.9793 | 0.9854 |
| 17 | 1058338 | 20375 | 1076896 | 4175 | 0.9923 | 0.9961 | 0.9811 | 0.9885 |
| 18 | 1175034 | 23536 | 1191684 | 9506 | 0.9926 | 0.9920 | 0.9804 | 0.9861 |
| 19 | 1177018 | 21552 | 1196463 | 4727 | 0.9928 | 0.9960 | 0.9820 | 0.9890 |
| 20 | 1293784 | 24643 | 1311294 | 10015 | 0.9933 | 0.9923 | 0.9813 | 0.9868 |
| 21 | 1295790 | 22637 | 1316031 | 5278 | 0.9932 | 0.9959 | 0.9828 | 0.9893 |
| 22 | 1412583 | 25701 | 1430862 | 10566 | 0.9933 | 0.9926 | 0.9821 | 0.9873 |
| 23 | 1414624 | 23660 | 1435606 | 5822 | 0.9935 | 0.9959 | 0.9835 | 0.9897 |
| 24 | 1531461 | 26680 | 1550452 | 11095 | 0.9937 | 0.9928 | 0.9829 | 0.9878 |
| cumulative | 19832125 | 423708 | 20161841 | 138270 | 0.9888 | 0.9931 | 0.9791 | 0.9860 |

Table 1: Statistics of CN Training

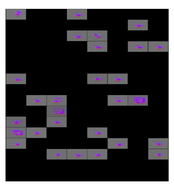

Figure 7: Real Image filter activation in CFFN Final Layer

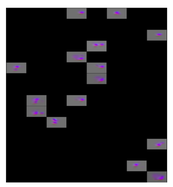

Figure 8: DCGAN Image filter activation in CFFN Final Layer

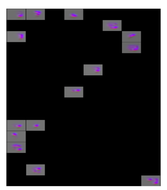

Figure 9: PGGAN Image filter activation in CFFN Final Layer

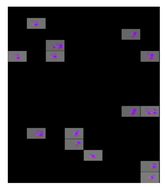

Figure 10: WGAN-CP Image filter activation in CFFN Final Layer

- Figures (6-9) indicate that many filters completely removed information, suggesting that the network identified relevant features when working with real and fake image pairs.
- Real images had more unique features than fake images, as shown by the higher number of nonzero filters (28) compared to GAN images (14-15).

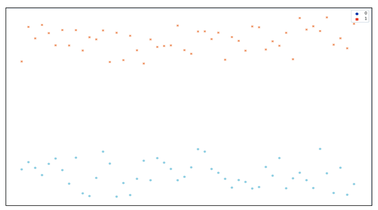

Figure 11: t-SNE of CFFN latent dimension with perplexity factor - 30

Figure 12: t-SNE of CFFN latent dimension with perplexity factor - 10

### Latent tensor visualization:

- Concatenate last three dense block outputs into 384-length tensors.
- Dimensionality reduction with PCA and t-SNE.

### Impact of perplexity on t-SNE:

- Perplexity of 30 is useful for visualizing separability of real and fake data.
- CFFN model learned discriminative features between (real, real) and (fake, fake) pairs.

## Setup: Final Experiments (Phase Two)

### Additional Dataset Generation:

- WGAN-GP and LSGAN models created additional fake image dataset with 40,000 images each.

| Method/Target | WGAN-GP | | DCGAN | | WGAN-CP | | LSGAN | | PGGAN | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Precision* | *Recall* | *Precision* | *Recall* | *Precision* | *Recall* | *Precision* | *Recall* | *Precision* | *Recall* |
| Method in [5] | 0.322 | 0.373 | 0.334 | 0.349 | 0.371 | 0.391 | 0.350 | 0.396 | 0.345 | 0.378 |
| Method in [6] | 0.769 | 0.602 | 0.749 | 0.689 | 0.809 | 0.743 | 0.808 | 0.761 | 0.817 | 0.703 |
| Method in [7] | 0.792 | 0.684 | 0.820 | 0.811 | 0.864 | 0.881 | 0.848 | 0.869 | 0.868 | 0.853 |
| Method in [8] | 0.830 | 0.671 | 0.827 | 0.796 | 0.882 | 0.869 | 0.862 | 0.854 | 0.881 | 0.875 |
| Method in [9] | 0.832 | 0.690 | 0.871 | 0.847 | 0.885 | 0.920 | 0.866 | 0.898 | 0.922 | 0.909 |
| Baseline-I [1] | 0.876 | 0.711 | 0.882 | 0.887 | 0.902 | 0.920 | 0.900 | 0.914 | 0.938 | 0.901 |
| Baseline-II [1] | 0.901 | 0.728 | 0.822 | 0.838 | 0.864 | 0.881 | 0.920 | 0.919 | 0.917 | 0.887 |
| Proposed [1] | 0.986 | 0.751 | 0.929 | 0.916 | 0.988 | 0.927 | 0.919 | 0.986 | 0.988 | 0.948 |
| Our network | 0.993 | 0.993 | 0.993 | 0.998 | 0.993 | 0.999 | 0.993 | 0.998 | 0.993 | 0.999 |
| **Our retrained** | **0.998** | **0.988** | **0.998** | **0.988** | **0.999** | **0.988** | **0.999** | **0.988** | **0.999** | **0.988** |

Table 2: Statistics of performance comparison of retrained model

### Retraining CFFN Model:

- Retrained CFFN model over 15 epochs and 88 batch size.
- Retrained model showed better precision for all GAN datasets.
- The recall was slightly lower indicating the model was classifying more real images as fake.

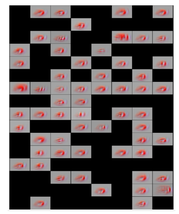

Figure 13: Real Image filter activation in retrained CFFN

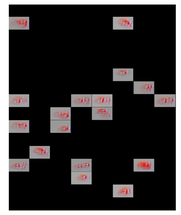

Figure 14: DCGAN filter activation in retrained CFFN

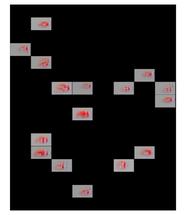

Figure 15: PGGAN filter activation in retrained CFFN

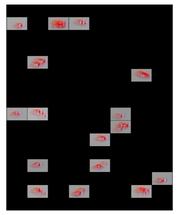

Figure 16: WGAN-CP filter activation on retrained CFFN

- Based on the figures (12-15), it can be inferred that the activations for fake images were similar to the previous training of CFFN, but in the retrained CFFN, the activation values were notably retained in the red channel.
- Furthermore, the retrained CFFN showed more activations in regards to real images, with twice as many activations observed in the output as compared to the previous training.
- The filter activations revealed that the retrained model was able to better distinguish features from real images when it was trained with additional data on fake images.

## Conclusions

- **Generalizability of the model:** Our model demonstrated strong generalizability by correctly identifying face images generated from multiple GAN architectures, including WGAN-GP, LSGAN, and CDCGAN, during the initial testing phase.
- **Improvement in accuracy with additional datasets:** By incorporating additional GAN datasets in the model's training, we observed a notable improvement in its ability to discern discriminative features within real images, leading to more accurate identification of fake and real images.

## Potential Future Improvements

- **Quantitative evaluation of GAN model:** By using various methods such as we can get a better understanding of the quality of our training data and the accuracy of our model based on different-rated fake images.
- **Visualizing model efficacy with distinct parameters**: We can setup additional datasets for testing our model by using available image generation models with different architectures.
- **Improving resource utilization and network performance:** We can explore impact of removing CFFN preprocessing on classification network, experiment with different values of "m" in the loss function, and try to reduce CFFN filters in the last layer or reduce the number of CFFN channels for improved resource utilization.